Survival guide for testing modern applications
Why modernize applications?

According to a Forrester survey, the primary business drivers behind application modernization efforts are business agility, innovation, and cost savings. In an age of immediate expectations, an agile organization that can change direction quickly is one that will enable business success in a fast-changing world. Innovation is the key to accelerating growth and providing competitive differentiation. For business applications, this can mean more mobile applications, more investment in the user experience, more embedded social tools, and frequent updates to stay competitive. And all this needs to be accomplished at a faster pace while optimizing operational processes to reduce costs.

To meet these business needs, the agile IT enterprise is adopting new technologies and architectures such as cloud computing, Web 2.0, composite applications, HTML 5, and mobile technologies as well as new approaches such as Agile, continuous delivery, DevOps, and other practices intended to accelerate delivery. These important IT initiatives can be big business disruptors; they also exacerbate the challenges of functional testing for the quality assurance (QA) organization.

Executive summary

In this paper you will learn how functional testing teams can modernize their practices to enable the business to stay agile and responsive to customer demands and competitive threats.

New application architectures, such as composite/hybrid applications, cloud computing networks, and mobile platforms introduce enormous challenges to today's functional testing teams, many of which are not equipped with the tools and knowledge to rise to the challenges. Iterative development processes such as Agile bring the need to test modern applications sooner, faster, and with less information available about the application; this requires an unprecedented level of cooperation among development, testing, and operations. Complex apps, increased platform coverage, and faster releases have led to a dire need for more investment in both manual and automated functional testing practices.

This paper will address these challenges in more detail, as well as outline ways your QA teams can invest to make key changes to allow the business to stay agile and competitive while releasing applications faster without sacrificing quality.

Best practices include starting the test process as early as possible, even before a completed application is ready to be tested, testing the service layer, looking at the big picture of what will need to be tested, at the end-to-end business process, and investing time up-front to design an effective yet change-ready test framework. Building a framework with reusable test components makes test creation and maintenance more efficient. Implement modern manual testing methods to eliminate the repetitive, error-prone tasks of manual testing and facilitate better communication. Find new opportunities to automate tests and simplify the transition from manual to automated testing. And for more thorough testing of composite applications, test the GUI layer, the non-GUI layer, and the complete end-to-end process using automation wherever possible.

IT initiatives = headaches for testing teams

The ongoing influx of new technologies and processes brings greater complexity and demands shorter testing cycles. An unprepared testing team can be seen as a roadblock to business agility and faster time to market. Here are some of the reasons why functional testing can be particularly difficult for these IT initiatives.

Agile: IT organizations are adopting Agile, continuous delivery, DevOps, and other practices aimed to speed up software delivery and minimize the impact of change. These practices increase automation, streamline cross-team handoffs, and allow IT to meet the business’ need for high quality at a fast pace of delivery. Unchanged functional testing methods will diminish the advantages of Agile methodologies. Testing needs to take place as early as possible in the lifecycle, often in parallel with development. How do you conduct testing during development to provide defect feedback before a sprint concludes (or without impacting the schedule)? How do you test before the GUI is available, or the composite application is even a complete system? How can you make your testers productive with minimal documentation on an application and still communicate comprehensive defect descriptions to facilitate fast remediation.

Without a conscious effort to improve testing practices, QA will fall behind and impact delivery schedules. Agile test teams must aggressively automate the testing for both the GUI and API layers, add value as early as possible in the lifecycle, learn to virtualize dependencies, conduct effective manual testing before automation is possible, and look for ways to integrate and kick off automated testing into development processes.
Composite applications: The QA organization must evolve its capacity to test modern applications with complex architectures that include rich GUIs, non-GUI business logic layers (aka API or headless layers), shared services (often SOA-based), and integrations with packaged applications. It is necessary to validate GUI and non-GUI functionality independently, and to visualize, validate, and report on the integrated test scenario for the entire business process that traverses the multiple layers of a composite application.

For example, how do you test a bank transaction that initiates a deposit, executes a database call, and sends a text to the customer while confirming back to the user that the deposit was successful? Too many of these processes won’t reveal their functionality through the GUI and must be validated another way. And if the service that sends a text to the customer is also used in a transaction that sends an overdraft notice, will there be additional defects that have gone unnoticed?

To ensure the same level of quality for a composite application equal to a siloed/self-contained application, testers must expand their capacity to test each layer of the application as well as the end-to-end business transactions. This is critical particularly if GUIs and APIs are shared by different applications and expose different subsets of functionality. It’s also critical to validate an entire business process through GUI and non-GUI layers or you risk releasing defects in backend processes not visible through the GUI. With so many components used by multiple applications, it can be difficult for the testing team to understand the dependencies between components themselves and the applications that use them, particularly when some of the services are unavailable or operate as black boxes provided by third parties.

Testing teams will quickly discover this is a tall order, and their existing tools and processes prevent them from testing composite applications effectively. Investment in smart automation and API level testing is key.

Mobile: Enterprise applications extending to mobile interfaces must be tested just as thoroughly as their PC counterparts. Users, whose expectations for quality have been set by their personal smart phones, have no patience for mobile enterprise applications that don’t behave properly; testing to ensure correct functionality is critical to productivity and customer loyalty. Mobile applications have all the testing complications of composite applications, but they don’t simply imitate the desktop environment. They have their own user interface requirements, business process flows, and infrastructure and hardware dependencies. And they are particularly challenging for the QA organization because there are so many possible devices to support. According to Forrester, “Your customers own many different devices that rely on different development environments...The multiple versions of the same OS, different screen sizes, and high number of devices mean that the costs of porting, maintaining, and promoting apps will remain high.” Manual testing can be extremely labor-intensive when working through smart phones, PDAs, tablets, and specialized equipment such as a rental car agent’s scanning and printing handheld device. It’s even worse when you consider the exponential permutations of data, usage scenarios, mobile network conditions, and carriers.

Because manual testing is too inefficient, and broad use case coverage is required, this is an area ripe for enhanced automated testing solutions.

How to modernize testing for an agile enterprise

QA organizations can address these issues and become full partners in implementing today’s modern applications as part of a truly responsive enterprise. They should engage in quality management efforts that result in more predictable delivery of applications with better customer satisfaction and fewer defects. They can accomplish this by employing processes and practices that allow testing early in the lifecycle, increasing automation wherever possible, leveraging frameworks to reduce time spent on test creation and maintenance, adding virtualized environments for testing unavailable services, leveraging accelerators and templates to cut down time spent on packaged application testing, and integrating to continuous integration systems for a unified and comprehensive quality validation process. As a result they’ll be able to test more, test faster, test sooner, and test continuously.
The HP approach to functional testing allows IT organizations to automate more, faster, and in a simple way.

- **Manual testing**—We recognize a large percentage of testing is still done manually. While appropriate for some situations, it often involves tedious test creation, high maintenance, and laborious test execution.
- **Mixed manual and automated testing**—Most organizations have a mix of manual and automated testing, conducted separately and often by different teams. Teams in this situation should look for more opportunities to automate existing manual test cases, as well as update their manual testing practices to improve output.
- **Frameworks**—To increase the velocity of your testing, we recommend utilizing a component-based framework; this involves creating test sets based on reusable components that are managed and shared. The result is reduced test maintenance, speedier test creation, and when utilized for both manual and automated tests, you improve your success rate of automating manual tests down the road.
- **Accelerated Frameworks**—After introducing a framework, test teams will quickly recognize that, like automation scripts, the framework requires time to create and maintain. By adding framework accelerators, they can significantly cut time spent on building and maintaining frameworks. Accelerators are templates with predefined frameworks and test cases that are built in and ready to run to test popular packaged applications.

HP Business Process Testing (HP BPT) software provides a framework for creating a library of reusable testing components that can be assembled into flows. It enables testing teams to increase application quality by decreasing the time needed for test creation and maintenance. And it lowers the cost of quality in the face of regular and rapid change.

Another advantage of taking the time up-front to design a test strategy with HP BPT is the ability to quickly determine which components to test manually and which to automate. It’s common for components to start out being tested manually and then transition to automation later if and when it makes sense to do so, but when taking full advantage of your tools, functionality may be automated from the start without much rework down the road. HP BPT also provides the added bonus of auto-generating test plan documentation and creating detailed linkages into HP Quality Center or HP ALM software requirements. Well-documented, well-managed tests continue to be useful through future test iterations, especially when tests change owners.

**For example, let’s say there are three business processes to be validated: create order, delete order, and update order. Once the business processes are modeled, it’s easy to identify duplication: all three use the same login function (see Figure 1). With a modular approach, the test component for the login function only needs to be created once, not three times, eliminating duplication of effort. If a change is made to the login function, it is only necessary to change the associated test component once, and all tests that use that component will be updated. This “test by composition” method can reduce test creation and maintenance effort. And this greater efficiency is realized whether the tests themselves are automated or performed manually.**

HP Automation Maturity Model

The HP approach to functional testing allows IT organizations to automate more, faster, and in a simple way.

- Manual testing—We recognize a large percentage of testing is still done manually. While appropriate for some situations, it often involves tedious test creation, high maintenance, and laborious test execution.
- Mixed manual and automated testing—Most organizations have a mix of manual and automated testing, conducted separately and often by different teams. Teams in this situation should look for more opportunities to automate existing manual test cases, as well as update their manual testing practices to improve output.
- Frameworks—To increase the velocity of your testing, we recommend utilizing a component-based framework; this involves creating test sets based on reusable components that are managed and shared. The result is reduced test maintenance, speedier test creation, and when utilized for both manual and automated tests, you improve your success rate of automating manual tests down the road.
- Accelerated Frameworks—After introducing a framework, test teams will quickly recognize that, like automation scripts, the framework requires time to create and maintain. By adding framework accelerators, they can significantly cut time spent on building and maintaining frameworks. Accelerators are templates with predefined frameworks and test cases that are built in and ready to run to test popular packaged applications.

The HP Functional Testing Suite promotes progress toward the most mature state of automation and allows organizations to simplify and expand their capacity for functional testing.

Here are specific steps the functional testing team can follow to modernize:

**1. Start with design:** Although this step is often neglected, taking the time to carefully design the test strategy pays off by making tests more efficient and easier to maintain. We recommend using a framework—creating a model of the business process and building a library of reusable test components that can be assembled into flows.
Accelerators—leverage templates to cut down time and costs

Even if you don’t utilize the features of HP BPT to develop a custom framework, you can still take advantage of a new set of accelerators designed for packaged application testing. Built on a foundation of HP Unified Functional Testing (UFT) and HP BPT for seamless integration, accelerators provide out-of-the-box test libraries for popular applications such as Oracle and SAP. The accelerators include fully automated test cases, a data-driven test data management system, and an automated business process reporting platform. Business can cut testing time for these environments by many magnitudes.

Remove bottlenecks with virtualized services

An important feature of the HP Functional Testing Suite is the ability to develop and test in constrained or incomplete environments. HP Service Virtualization is designed to quickly establish and provide access to limited or unavailable services through a simulated environment. Scenarios where this is important include:

1. Running a service that incurs a cost such as credit card processing
2. The service is not available or broken
3. The service is incomplete or under construction
4. Doing a regression testing to validate that a changed service didn’t introduce unexpected behavior

In UFT, you can load one or more simulation projects and displays if and when the service was deployed, plus information about the virtualization server. Once deployed, calls to the virtualized service can be integrated into steps within a UFT automated test. HP Service Virtualization enables you to move forward with functionality that is dependent on an unavailable service, removing roadblocks to effective testing.

Figure 3
HP UFT graphical user interface

2. Increase and improve automation: Increasing automation will help the QA organization to test early in the lifecycle, to increase capacity without increasing resources, to adapt faster to a constantly changing environment, and catch up with the fast pace of modern development practices.

HP Unified Functional Testing (UFT) includes the well-known HP QuickTest Professional GUI test automation software, along with HP Service Test and HP Service Test Management software for advanced services testing and management capabilities.

Via a modern user interface, the latest HP UFT provides shortcuts and functionality aligned with market-leading developer IDEs, but customized for testing, so that both developers and less technical testers will be productive in a very short amount of time. Both API tests and GUI tests are displayed in a graphical view called the “canvas,” which provides a visual representation of the test flow. In the canvas, you can manage actions, change their order in the test, and run and debug tests with drag-and-drop capabilities—features that facilitate test design, increase productivity, and allow non-technical users to even participate in highly technical aspects of testing through APIs. (see figure 4) The UI also offers enhanced authoring and editing capabilities such as auto-completion of code, customized and built-in code snippets, and other activities such as file-content checkpoints and bitmap checkpoints. And with the integration to HP Quality Center or HP ALM, users can utilize workflows to define actions and conditions that simplify their work and allow sharing of assets across projects.

UFT has also expanded its broad array of technology testing by adding out-of-the-box support for Qt applications, Adobe Flex, JSON, IDocs, and RFCs for SAP systems.
Image-based object recognition automation

Rich Internet Application (RIA) and Web 2.0 technologies are needed to provide a satisfying user experience, but bring their own unique testing challenges. Functional testing tools must be able to test a diverse, ever-expanding set of toolkits, such as those that leverage Microsoft® Silverlight®, and the wide range of Ajax offerings including Dojo Toolkit, Yahoo! User Interface (YUI), ASP.NET AJAX, and Google® Web Toolkit (GWT), to name a few. While most do not change the underlying web infrastructure, each toolkit has different web controls with unique behaviors, which may not be automatically recognized by testing tools. HP UFT Insight, part of the latest release, allows identification of controls by their appearance only, regardless of their native properties. This image-based identification allows testing teams to quickly adapt to fast-changing environments and toolkit releases.

Integration to CI systems

Also new with the latest release, UFT integrates with the Jenkins and Hudson continuous integration systems (both open source CI systems, and provides full flexibility to tailor methodologies to how the organization works. A continuous integration setup will provide immediate test results without a dependency on the QA team—a critical approach when working within Agile sprints or iterations. Configuring UFT test scripts to run automatically as part of nightly builds together with other developer supplied tests results in a more robust quality validation process that includes unit, regression, integration, and smoke tests. See figure 5.
3. Modernize manual testing: While organizations are increasingly trying to leverage automated testing, manual testing still accounts for about 75% of all functional testing. However, testers can adopt modern manual-testing methods to simplify and minimize tedious and error-prone tasks. Instead of typing in data from spreadsheets, taking notes during test runs, typing results into the testing suite, and repeating these steps over and over again for different environments and configurations, they should seek out a solution to streamline these steps.

HP Sprinter manual testing solution simplifies and expedites manual testing while increasing cross-team collaboration. HP Sprinter automatically logs test cases directly from HP Quality Center or HP ALM and presents them on the screen. To minimize errors, HP Sprinter automatically injects data from an external spreadsheet to the application under test. HP Sprinter automatically logs test cases from HP Quality Center or HP Application Lifecycle Management (HP ALM) software and presents them on the “application under test” screen. From here, the user can view test steps, open a defect, add comments, and log actual results.

HP Sprinter automatically logs all user actions in three formats: a textual description of all user actions (see Figure 3), a screen capture of all user actions, and a video recording of the entire test execution. Testers can attach this detailed description to a new defect with the click of a button. This dramatically improves defect communication to developers to facilitate fast remediation.

For applications that are primarily tested manually (including mobile applications) HP Sprinter provides “mirror testing” by cloning a test multiple times for different environments and configurations. This enables the tester to run a test only once while HP Sprinter replicates it on multiple machines at the same time. It has been shown to boost overall manual test productivity by 50% to 70%.3

With HP Sprinter, HP BPT, and HP UFT, test results and all relevant documentation are automatically saved to HP ALM or HP Quality Center, providing testers as well as other stakeholders with full traceability of test results to test cases, requirements, and defects in a single system of record.
The functional testing journey

In order for QA teams to address emerging market needs including the adoption of cloud computing, Web 2.0, mobile, and composite applications, there must be a transformation in the way functional testing is done. Successful functional testing teams start the testing process early in the application lifecycle. Even before an application is ready to be tested, they look at the testing problem holistically and take time to design their test strategy. The strategy includes the use of a framework and the creation of reusable test components. They engage with the business and incorporate testing early in the lifecycle with exploratory testing and modern manual-testing methods. And they begin testing the headless layer even before a GUI becomes available. Forward-looking QA teams look for new opportunities to automate testing when and where it makes sense to do so. And for the areas where they continue to test manually, they accelerate the process by automating the tedious, repetitive tasks to cut testing time. And finally, they fully validate composite applications by testing the GUI layer, the headless layer, and the complete end-to-end process.

For more customer feedback go to techvalidate.com/portals/hp-unified-functional-testing-software-customer-proof-points.

The HP Functional Testing Suite—HP UFT, BPT, BPT Accelerators, Sprinter, and Service Virtualization—is an integrated part of the overall HP testing experience that allows customers to evolve their testing solution to match their changing and growing needs. Moreover, it allows testing teams to effectively provide testing services in Agile development environments and respond to a fast-changing environment without sacrificing application quality and innovation. With this toolset together with mobile testing, they can test more, test faster, and test continuously.

To learn more about HP functional testing solutions and download trial versions of the software, visit hp.com/go/functionaltesting.

---

1 Base: 206 IT decision makers with insight into budgeted modernization activities. Source: Clearing Your Path to Modern Applications and Business Agility, Forrester Research, April 6, 2010. A Forrester Consulting thought leadership paper commissioned by HP.

2 Forrester, 2011 Mobile Trends, Thomas Husson and Julie A. Ask, January 24, 2011

3 Customer testimonials are available at techvalidate: see techvalidate.com/portals/hp-sprinter

---

Get connected
hp.com/go/getconnected

Get the insider view on tech trends, support alerts, and HP solutions.

© Copyright 2011–2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft and Silverlight are U.S. registered trademarks of Microsoft Corporation. Google is a trademark of Google Inc. Java and Oracle are registered trademarks of Oracle Corporation and/or its affiliates.

4AA3-5237ENW, Created June 2011; Updated December 2012, Rev. 2